

WHAT IS CLAIMED IS:

- Sub B-1
1. A method for processing an input file in a file system, wherein the input file has an input file name, comprising:
 - applying a function to map the input file name to a value; and
 - processing a data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name.
 2. The method of claim 1, wherein the mapped-to values require fewer bits of storage than the file names.
 3. The method of claim 1, wherein the function is a hash function that maps the input file name to an integer value, and wherein the data structure includes an entry for each possible integer value capable of being generated from the hash function.
 4. The method of claim 3, wherein processing the data structure to determine whether there is a preexisting file comprises determining whether the entry for the integer value to which the input file name maps indicates the presence of one preexisting file mapping to the same integer value as the input file name.
 5. The method of claim 4, wherein the data structure is a one-dimensional array and wherein each entry is capable of having one of two values, further comprising setting the entry to a first value if there is one preexisting file name in the file system that maps to the integer value for the entry, and wherein determining whether there is one preexisting file comprises determining whether the entry for the integer value to which the input file name maps has the first value.

09409613 100199

1 6. The method of claim 1, further comprising:
2 applying the function to each file name in the file system to map each file
3 name to one value; and
4 indicating in the data structure, for each file name, that there is one preexisting
5 file for the value to which the file name maps.

1 7. The method of claim 6, wherein the input file is the subject of an
2 access request, further comprising scanning each file in the file system to determine if
3 there is at least one preexisting file having the same name as the input file name if
4 there is one preexisting file in the file system having a name that maps, according to
5 the function, to the same value to which the input file name maps.

1 8. The method of claim 7, wherein the access request to the input file is
2 to add the input file as a new file to the file system, further comprising:
3 adding the input file as a new file to the file system if no preexisting file in the
4 file system has the same name as the input file name; and
5 rejecting the access request if there is a preexisting file in the file system
6 having the same name.

1 9. The method of claim 7, wherein the access request to the input file is
2 to update a file in the file system with data from the input file, further comprising:
3 updating a preexisting file in the file system having the same name as the
4 input file with the data in the input file if there is such a preexisting file; and
5 rejecting the access request if there is no preexisting file in the file system
6 having the same name as the input file name.

09409613-100199

1 10. A system for processing an input file in a file system, wherein the
2 input file has an input file name, comprising:
3 means for applying a function to map the input file name to a value; and
4 means for processing a data structure to determine whether there is a
5 preexisting file in the file system having a name that maps, according to the function,
6 to the same value to which the input file name maps, wherein two files that map to a
7 same value according to the function are capable of having a same name.

1 11. The system of claim 10, wherein the mapped-to values require fewer
2 bits of storage than the file names.

1 12. The system of claim 10, wherein the function is a hash function that
2 maps the input file name to an integer value, and wherein the data structure includes
3 an entry for each possible integer value capable of being generated from the hash
4 function.

1 13. The system of claim 10, wherein the means for processing the data
2 structure to determine whether there is a preexisting file comprises determining
3 whether the entry for the integer value to which the input file name maps indicates the
4 presence of one preexisting file mapping to the same integer value as the input file
5 name.

1 14. The system of claim 13, wherein the data structure is a one-
2 dimensional array and wherein each entry is capable of having one of two values,
3 further comprising setting the entry to a first value if there is one preexisting file name
4 in the file system that maps to the integer value for the entry, and wherein determining
5 whether there is one preexisting file comprises determining whether the entry for the
6 integer value to which the input file name maps has the first value.

09409613-100199

1 15. The system of claim 10, further comprising:
2 means for applying the function to each file name in the file system to map
3 each file name to one value; and
4 means for indicating in the data structure, for each file name, that there is one
5 preexisting file for the value to which the file name maps.

1 16. The system of claim 10, wherein the input file is the subject of an
2 access request, further comprising means for scanning each file in the file system to
3 determine if there is at least one preexisting file having the same name as the input
4 file name if there is one preexisting file in the file system having a name that maps,
5 according to the function, to the same value to which the input file name maps.

1 17. The system of claim 16, wherein the access request to the input file is
2 to add the input file as a new file to the file system, further comprising:
3 means for adding the input file as a new file to the file system if no preexisting
4 file in the file system has the same name as the input file name; and
5 means for rejecting the access request if there is a preexisting file in the file
6 system having the same name.

1 18. The system of claim 16, wherein the access request to the input file is
2 to update a file in the file system, with data from the input file, further comprising:
3 means for updating a preexisting file in the file system having the same name
4 as the input file with the data in the input file if there is such a preexisting file; and
5 means for rejecting the access request if there is no preexisting file in the file
6 system having the same name as the input file name.

1 19. An article of manufacture for processing an input file in a file system,
2 wherein the input file has an input file name, the article of manufacture comprising

09409613-100199

3 computer usable media including at least one computer program embedded therein
4 that causes the computer to perform:
5 applying a function to map the input file name to a value; and
6 processing a data structure to determine whether there is a preexisting file in
7 the file system having a name that maps, according to the function, to the same value
8 to which the input file name maps, wherein two files that map to a same value
9 according to the function are capable of having a same name.

1 20. The article of manufacture of claim 19, wherein the mapped-to values
2 require fewer bits of storage than the file names.

1 21. The article of manufacture of claim 19, wherein the function is a hash
2 function that maps the input file name to an integer value, and wherein the data
3 structure includes an entry for each possible integer value capable of being generated
4 from the hash function.

1 22. The article of manufacture of claim 21, wherein processing the data
2 structure to determine whether there is a preexisting file comprises determining
3 whether the entry for the integer value to which the input file name maps indicates the
4 presence of one preexisting file mapping to the same integer value as the input file
5 name.

1 23. The article of manufacture of claim 22, wherein the data structure is a
2 one-dimensional array and wherein each entry is capable of having one of two values,
3 further comprising setting the entry to a first value if there is one preexisting file name
4 in the file system that maps to the integer value for the entry, and wherein determining
5 whether there is one preexisting file comprises determining whether the entry for the
6 integer value to which the input file name maps has the first value.

05409613-100199

1 24. The article of manufacture of claim 19, further comprising:
2 applying the function to each file name in the file system to map each file
3 name to one value; and
4 indicating in the data structure, for each file name, that there is one preexisting
5 file for the value to which the file name maps.

1 25. The article of manufacture of claim 24, wherein the input file is the
2 subject of an access request, further comprising scanning each file in the file system
3 to determine if there is at least one preexisting file having the same name as the input
4 file name if there is one preexisting file in the file system having a name that maps,
5 according to the function, to the same value to which the input file name maps.

1 26. The article of manufacture of claim 25, wherein the access request to
2 the input file is to add the input file as a new file to the file system, further
3 comprising:
4 adding the input file as a new file to the file system if no preexisting file in the
5 file system has the same name as the input file name; and
6 rejecting the access request if there is a preexisting file in the file system
7 having the same name.

1 27. The article of manufacture of claim 25, wherein the access request to
2 the input file is to update a file in the file system with data from the input file, further
3 comprising:
4 updating a preexisting file in the file system having the same name as the
5 input file with the data in the input file if there is such a preexisting file; and
6 rejecting the access request if there is no preexisting file in the file system
7 having the same name as the input file name.

add
D2

09409613-100199